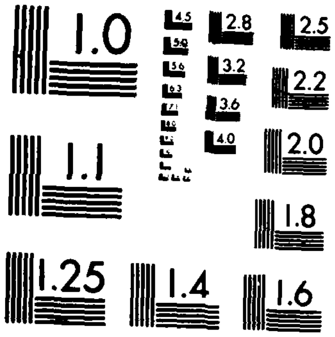
AD-A174 393 ROBUST COARTICULATORY MODELING FOR CONTINUOUS SPEECH RECOGNITION(U) BBN LABS INC CAMBRIDGE MA R SCHWARTZ ET AL OCT 86 BBN-6383 1/1

UNCLASSIFIED F/G 5/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

BBN Laboratories Incorporated

A Subsidiary of Bolt Beranek and Newman Inc.

(12)

BBN Report No. 6383

Mid-Term Technical Report

ROBUST COARTICULATORY MODELING FOR CONTINUOUS SPEECH RECOGNITION

R. Schwartz, Y-L. Chow, M.O. Dunham, O.Kimball,
M. Krasner, F. Kubala, J. Makhoul, P. Price, S. Roucos

October 1986

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
BBN Report No. 6383

2. GOVT ACCESSION NO.
AD-A174393

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Robust Coarticulatory Modeling for Continuous Speech Recognition

5. TYPE OF REPORT & PERIOD COVERED
Midterm Technical Report
Jan. 1985 - Sept. 1986

6. PERFORMING ORG. REPORT NUMBER
BBN Report No. 6383

7. AUTHOR(s)
R. Schwartz, Y-L. Chow, M.O. Dunham, O. Kimball, M. Krasner, F. Kubala, J. Makhoul, P. Price, S. Roucos

8. CONTRACT OR GRANT NUMBER(s)
N00014-85-C-0279

9. PERFORMING ORGANIZATION NAME AND ADDRESS
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Department of the Navy
Arlington, Virginia 22217-5000

12. REPORT DATE
October 1986

13. NUMBER OF PAGES
35

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of the document is unlimited. It may be released to the Clearinghouse, Dept. of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

speech recognition, phonetic recognition, continuous speech, hidden Markov model, coarticulation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We report on our progress to date in developing phonetic models that are appropriate for large-vocabulary, continuous speech recognition. A simple but powerful formalism, hidden Markov modeling, is used for modeling phonemes in context. Using robust automatic training methods, the model parameters are estimated from a given amount of speech training data in a maximally effective way. Efficient search strategies are then

20. Abstract (continued)

utilized for performing phonetic recognition in continuous speech. Performance results are given in terms of phonetic accuracy and word accuracy for several problems of interest in continuous speech recognition.

BBN Report No. 6383

ARPA Order Number 4707
Contract Number N00014-85-C-0279
Contract Duration: 17 Jan 1985 - 16 Jan 1988
Principal Investigator: J. Makhoul, (617)497-3332

Mid-Term Technical Report

ROBUST COARTICULATORY MODELING FOR CONTINUOUS SPEECH RECOGNITION

R. Schwartz, Y-L. Chow, M.O. Dunham, O.Kimball,
M. Krasner, F. Kubala, J. Makhoul, P. Price, S. Roucos

October 1986

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Table of Contents

List of Figures

List of Tables

1. Executive Summary

The purpose of this project is to perform research into algorithms for the automatic recognition of individual sounds or phonemes in continuous speech. The algorithms developed should be appropriate for understanding large-vocabulary continuous speech input and are to be made available to the Strategic Computing Program for incorporation in a complete word recognition system.

This report describes ~~our~~ progress to date in developing phonetic models that are appropriate for continuous speech recognition. In continous speech, the acoustic realization of each phoneme depends heavily on the preceding and following phonemes: a process known as *coarticulation*. Thus, while there are relatively few phonemes in English (on the order of fifty or so), the number of possible different acoustic realizations is in the thousands. Therefore, to develop high-accuracy recognition algorithms, one may need to develop literally thousands of relatively distinct phonetic models to represent the various phonetic contexts adequately. Developing a large number of models usually necessitates having a large amount of speech to provide reliable estimates of the model parameters. The major contributions of ~~our~~ work ~~reported here~~ are the development of:

- A simple but powerful formalism for modeling phonemes in context.
- Robust training methods for the reliable estimation of model parameters by utilizing the available speech training data in a maximally effective way.
- Efficient search strategies for phonetic recognition in continuous speech, which minimize the time needed for recognition while maintaining high recognition accuracy.

We believe that the results achieved thus far have been unmatched in the recognition of large-vocabulary, continuous speech.

We model each of our phonemes in each phonetic context by a simple three-state Hidden Markov Model (HMM). (A HMM is a Markov chain where the "output" symbols are probability distributions instead of single symbols.) HMMs have been demonstrated to be very effective in modeling speech variabilities in time and in frequency. The parameters of each HMM are estimated from a given amount of speech (reserved for training the model parameters) via an automatic training algorithm that requires little or no human interaction. The results of the training is a set of phonetic models that are combined in a robust manner and are used in an efficient search strategy for the recognition of continuous speech.

To measure the phonetic accuracy of our recognition algorithm, we tested it on continuous speech from a single speaker with no restrictions on vocabulary. The result was 81% phonetic accuracy (i.e., 19% of the phonemes were either recognized incorrectly or were missed by the algorithm) and 12% additional phonemes were inserted. The 81% phonetic accuracy using our context-dependent models should be compared with a phonetic accuracy of only 62% when context is not utilized.

While phonetic recognition accuracy is a reasonable measure of our phonetic recognition algorithm, it is not a meaningful measure for most speech recognition applications where word accuracy is a more desirable measure. So, we incorporated our phonetic recognition algorithm in a continuous word recognition system, developed under a separate effort, and measured the resulting word recognition accuracy. Using continuous sentences spoken from a 334-word electronic mail task, the system achieved 90% word accuracy when all 334 words were allowed at every point in the sentence. However, when a grammar appropriate for the electronic mail task was used by the system, the word accuracy increased to 98.8%, averaged over three male and one female speakers.

2. Introduction

Hidden Markov Models (HMMs) have been shown to provide an effective statistical formalism for speech recognition. They have been used to model whole words in both isolated [1] and continuous [2] speech recognition. They have also been used to model phonemes for continuous speech recognition [2, 3, 4]. The Hidden Markov Model has two important advantages over many other models for speech. First, it provides a well-defined structural model for variability in both time and in frequency (spectral variation), both of which occur in speech. Second, once the structure of the models are specified the parameters of the models can be estimated automatically with a large amount of speech data using the forward-backward or Baum-Welch algorithm [5].

It is generally assumed that large vocabulary continuous speech recognition systems should be phonetically based. That is, each word in the lexicon is decomposed into phoneme subunits, each of which is modeled separately. The use of a phonetic model makes it easy to model phonological variation both within and between words. It also makes it possible for a new speaker to use the system without first saying all the words in the lexicon.

If the basic unit used to model speech represents the phoneme, it is implicitly assumed that the acoustic realization of each phoneme is independent of the phonetic context (e.g., the word) in which it is spoken. However, we know that phonemes are affected significantly - particularly near the transitions - by the adjacent phonemes. A statistical model like the HMM, can account for this effect with a large variance or, in the case of a discrete model, a greater number of allowed spectra in the transition regions than in the steady-state regions of the phoneme. However, this unconditioned relaxation of the model does not correctly reflect the conditional relation between each phoneme and its neighbor.

A common approach to deal with this problem is to utilize speech units larger than phonemes, such as diphones [6, 7], demisyllables [8], syllables [9, 10] etc. When larger acoustic units are used, the acoustic realization of the interior parts (far from neighboring phonetic contexts) will be largely independent of any neighboring units. However, both ends are still affected considerably by neighboring phonemes. Thus the problem is only partially solved. Furthermore, there are many more of these larger units than there are phonemes. For instance, there are about 2,500 diphones and about 10,000 syllables in English. Since most of these diphones or syllables will not occur even once in a data base of reasonable size it is impossible to gather detailed statistics about the likely acoustic realization of all of them from such a data base.

In this paper we present a more consistent formulation and solution for dealing with phonetic coarticulation and the problem of training large numbers of statistical models. We also report results for a series of experiments that demonstrate the usefulness of this new model. The framework for modeling coarticulation is developed in Section 3. Section 4 describes an experiment on the "E-set problem", a canonical illustration of the issues of phonetic coarticulation, minimal pair distinctions, and training set size. In Section 5, we describe experiments with continuous phonetic recognition. The methods were extended to continuous word recognition, and word-dependent coarticulatory effects. These experiments are described in Section 6. Section 7 contains a similar experiment for continuous speech using a grammar. Finally, brief conclusions are made in Section 8.

3. Framework for Modeling Coarticulation

To explain our model for coarticulation, we must first define some terms. Figure 1 illustrates several different levels of representation for speech.

The figure illustrates the levels of words, phonemes, allophones, allophone models, and analyzed speech parameters. The phrase shown is "grey whales". The purpose of speech recognition is to determine the sequence of words corresponding to an observed utterance. We often decompose words into sequences of basic speech sounds or phonemes, to try to reduce the problem of modelling many words to the problem of modelling a smaller number of units. We observe that these basic units exhibit systematic acoustic variation as a function of their phonetic environment. To capture this systematic variation we must first define context-dependent allophones or variants of each phoneme. An allophone is defined as any variant of a phoneme, which may be statistically different from other allophones of that phoneme. We have shown allophones defined by the preceding and following contexts. We will often use the terms "left" and "right" instead of "preceding" and "following". At the bottom of the figure is shown a schematic of the formant tracks corresponding to a single utterance of the phrase. This (or any other) parametric representation of the spoken speech will be different for each utterance of the phrase. Therefore, we need statistical models to represent the likely acoustic realizations (phones) for each allophone. While many different statistical models are available, we have chosen to use hidden Markov models as our basic allophone model for the reasons given previously.

As illustrated in the figure, the coarticulation effects bridge all the phoneme boundaries. Thus, using a larger unit such as the diphone, demisyllable or syllable ignores the contextual effects at their boundaries. If, instead, we allow the definition of the speech unit to extend beyond its duration, then we can model any amount of dependency that we wish. For example, in the illustration shown, we have modeled the effect of each phoneme on its immediate neighbors. This idea of *context-dependent* units is key to the modeling of coarticulation. Next, we discuss the issues of training set size and robustness that arise with the use of large numbers of models.

Training Problem

For any units with context-dependency larger than the phoneme, we will have a training problem. While some of the contexts may occur frequently, many will not occur with sufficient frequency to estimate a robust acoustic model. In fact, large numbers of the possible contexts will not occur at all in any particular set of training speech.

COARTICULATION MODEL

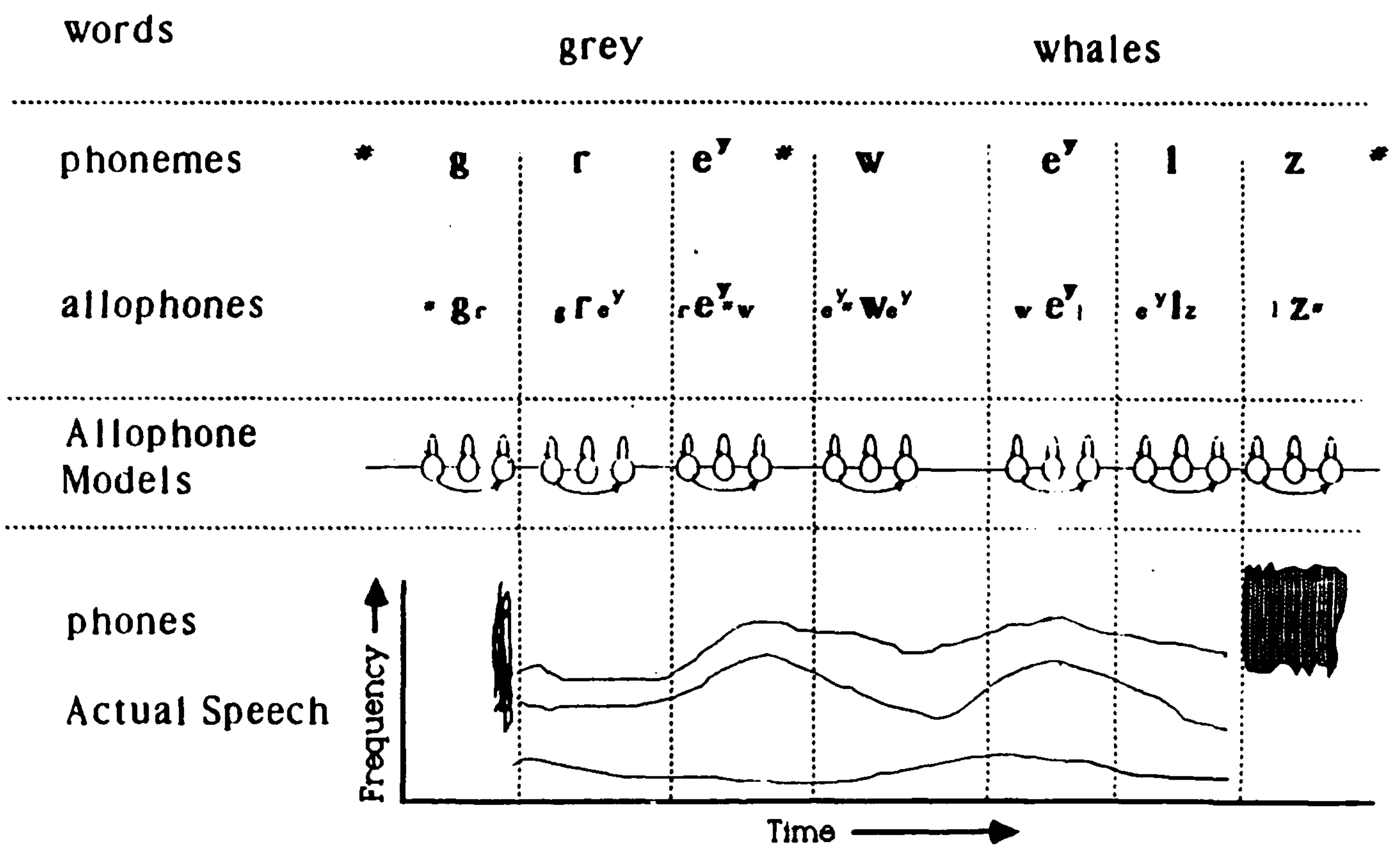


Figure 1: Several Different Levels of Representation for Speech. The phrase is "grey whales".

A simple solution would be to use the most complex context-dependent model with a sufficient number of training samples. For example, let us say we want a model for the /e^y/ in "whales". If the word whales has appeared a few times, we would use the model of /e^y/ that depended on the word whales. If not, we might use a model of all /e^y/ that are preceded by /w/ and followed by /l/ (as in "away late"). If this context did not occur, then we could fall back to a model dependent on the left or right context alone (as in "wait" or "tail"). Or if nothing else, we could resort to the context-independent model derived from all /e^y/ phoneme tokens. This algorithm for choosing the model, however, does not make optimal use of the training data, and does not properly account for coarticulatory phenomena. To solve this problem, we must examine more closely how coarticulation interacts with our model for a phoneme.

Figure 2 illustrates the HMM that we use to model a phoneme. The circles represent states of the model. We define s_t to be the state of the Markov process at time t. At each time, t, we also have an observed spectral envelope model, expressed as a vector, $\underline{x}_t$. With each state, i, is associated a probability density function

$$b_i(\underline{x}) = p(\underline{x}_t | s_t = i); \qquad i=1,2,3 \tag{1}$$

for the observed spectral vector, $\underline{x}_t$, given that the process is in state i at time t. Since the process is Markov, the probability densities do not depend on t. In our implementation, we use discrete probability densities for the vector $\underline{x}$. First, a portion of the training speech for a speaker is analyzed and then used to determine a codebook of spectral templates using a clustering algorithm [11]. Then, for any spectral envelope model vector, $\underline{x}$, we search the codebook for the template vector that is closest (Vector Quantization). The index of the closest vector, v_t, (refered to below as the "VQ spectrum") then defines a bin of a discrete probability density. The probability densities in our HMMs, then, have a probability for each of the possible bins:

$$b_i(v) = p(v_t = k | s_t = i); \qquad i=1,2,3; \qquad \mathit{for\ \ all\ \ k} \tag{2}$$

For each allowed transition (indicated by the arrows in Figure 2) we have a transition probability

$$a_{ij} = P\,(s_t = j | s_{t-1} = i) \tag{3}$$

the probability of state i being followed by state j. While the relation is not direct, we find it useful to think of the states as corresponding to the beginning, middle, and end of a phoneme.

Both experience and reason tell us that the coarticulatory effect of an adjacent phoneme is greatest in the part of the phoneme closest to that adjacent phoneme. For example, a phoneme to the left will have the most effect on the left part (state 1) of a phoneme, and the least effect on the right part (state 3). To account for both the nature of coarticulation and the requirements for robust statistical models, we use a *combined context model* as shown in the following example:

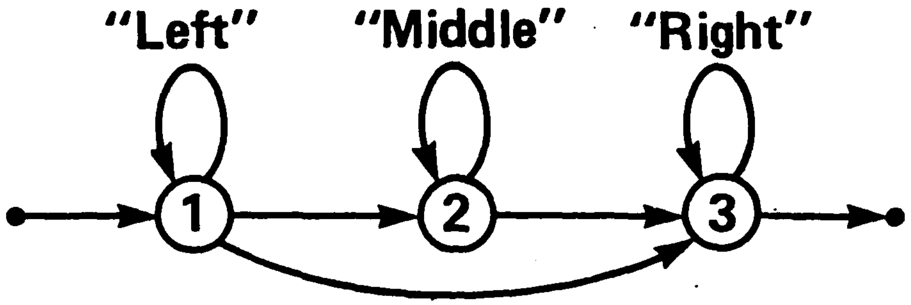


Figure 2: Hidden Markov Model of a Phoneme. States 1, 2, and 3 are assumed to correspond approximately to the left, middle, and right portion of a phoneme.

Example:
Model for /e^y/ in "whales"

$$\begin{aligned}\hat{p}(\underline{x}|e^y \text{ in whales}) = \; & \lambda_1 \; p(\underline{x}|w\, e^y\, l) \\ & + \lambda_2 \; p(\underline{x}|w\, e^y\,) \\ & + \lambda_3 \; p(\underline{x}|\, e^y\, l) \\ & + \lambda_4 \; p(\underline{x}|\, e^y\,) \\ & + \lambda_5 \; p(\underline{x}|\, e^y \text{ in whales})\end{aligned}$$

$$\underline{\lambda} = f(\#\ \text{Occurrences},\ \text{State})$$

$$\sum_{k=1}^{5} \lambda_k = 1$$

That is, the combined model, $\hat{p}$, is a linear combination of the various context-dependent models. The weight vector, $\underline{\lambda}$, depends on the state of the phoneme model (left, middle, right), and the amount of training for each model.

Figure 3a shows the basic HMM model for a phoneme. In Figure 3b, the second state and all its transitions are replaced by a *subnetwork* of three different context models in parallel, with weights to combine them. For the case of /e^y/ in *whales*, we define some terms below:

$$b_2L = p(v_t|s_t = 2, we^y) \tag{4}$$
$$\lambda_2 L = \mathit{weight\ of\ left\text{–}context\ model\ for\ state}\ 2\ \mathit{which\ is\ a\ function\ of}\ N_{we^y}$$

where N_c is the number of occurrences of context c in the training data. The sum of the weights, λ, from any node is 1. During forward-backward training the models are kept separate. Prior to recognition, the models for a state can be combined into a single probability density to save computation. Thus, during recognition, the HMM is of the same complexity as a single unconditioned model.

Finally, to demonstrate the behavior of the context-dependent models, we illustrate the actual HMM models in Figure 4. Figure 4 shows the various context-dependent models for the /EY/ in "WAY". The models shown, from top to bottom are context-independent (phoneme), left-context, right-context, and the combined model. From left to right are the three states of the model. Each pdf is shown as the probability of each VQ spectrum, from 1 to 256. On the left of the figure, we show, for each model, the number of occurrences of that model in the training set. As we examine the pdfs of the combined model, we see that the left pdf (state 1) of the combined model is most similar to the left pdf of the left-context model, the middle pdf is most like the context-independent pdf, and the right pdf is most like the right-context pdf. Each of the context-dependent pdfs is smoothed somewhat by the context-independent pdf, since the amount of training is not large.

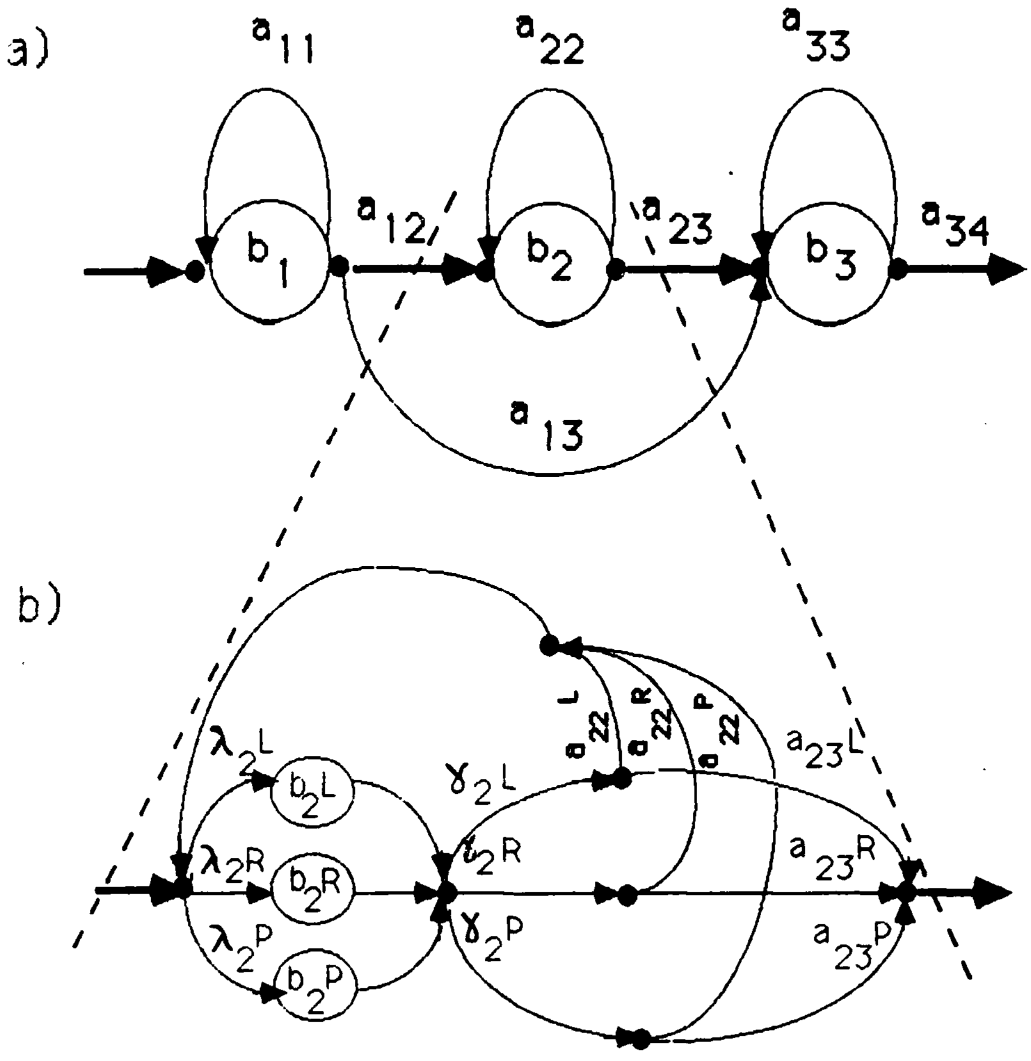
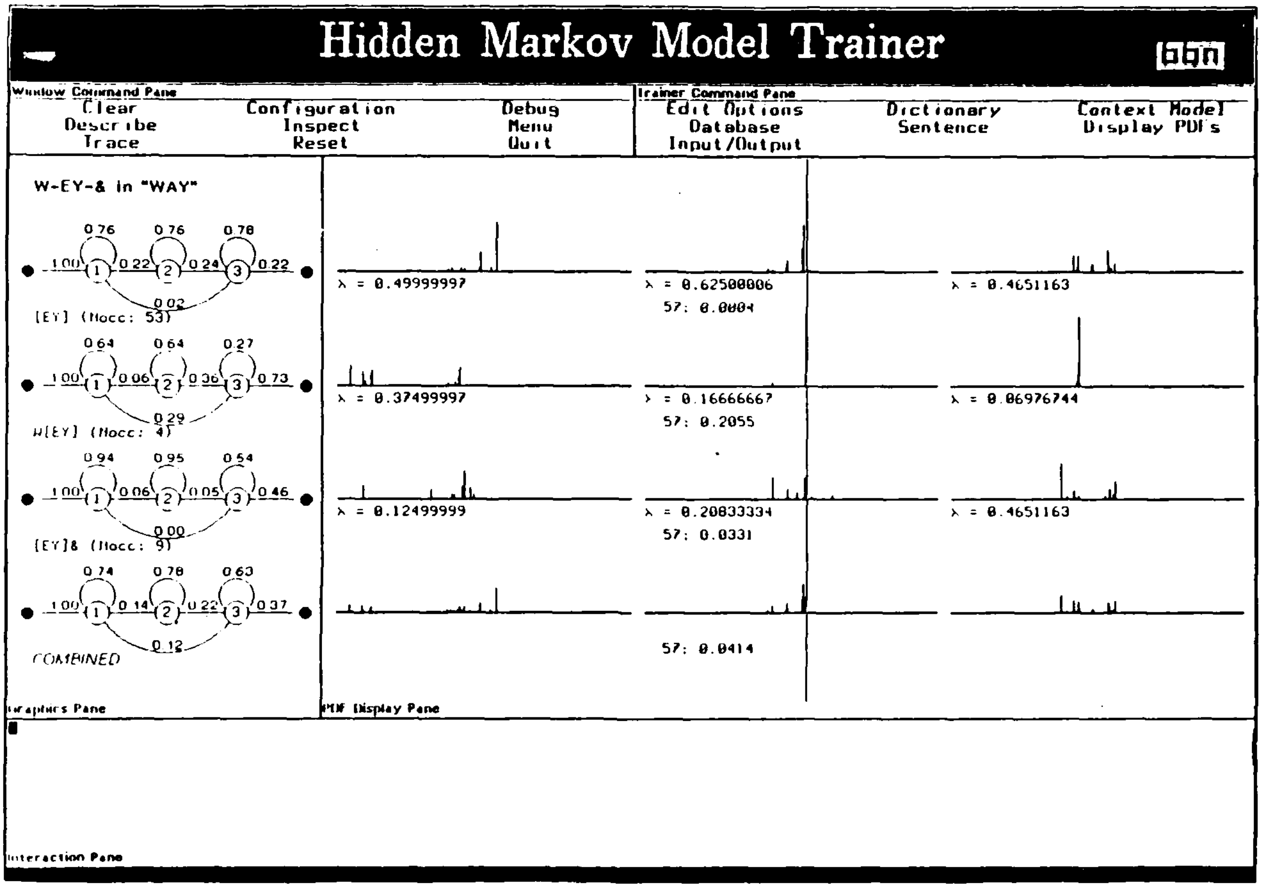


Figure 3: Expanded Hidden Markov Model. a) Model of a phoneme b) Expanded model for state 2.



```
λ = 0.49999997        λ = 0.62500006         λ = 0.4651163
                        57: 0.0004

λ = 0.37499997        λ = 0.16666667         λ = 0.06976744
                        57: 0.2055

λ = 0.12499999        λ = 0.20833334         λ = 0.4651163
                        57: 0.0331

                        57: 0.0414
```

Figure 4: Spectral pdfs for 3 states for the phoneme /EY/ in "WAY". Models shown from top to bottom are context-independent, left-context, right context, and the combined model.

To summarize, we have argued that we can model coarticulation effects by the use of context-dependent models of phonemes. Furthermore, to avoid the lack of robustness due to insufficient amounts of training, we can smooth these detailed context-dependent models with well trained context-independent models. The amount of the smoothing depends on both the location of the HMM state in the phoneme, and the amount of training available for that particular context. In the following four sections we describe a succession of experiments designed to demonstrate the effectiveness of the coarticulation model proposed above.

4. E-set Problem

The "E-set" is the set of nine letters of the English alphabet that rhyme with E. They are B, C, D, E, G, P, T, V, Z. They provide a few interesting problems for speech recognition. First, since they differ phonetically in only one phoneme, they require minimal pair distinctions. Second, since most of the duration of each utterance is the /i/ phoneme, one has to be careful that random statistical variation in this region does not dominate in the total discrimination score. Third, the models for the consonants do not depend on phonetic context, since they always appear preceded by silence (in isolated speech), and followed by /i/. The /i/ phoneme, however, appears with 9 different left contexts.

A single speaker said each of the 9 letters in isolation. The set of 9 letters was repeated 40 times in randomized sequences. The speech was lowpass filtered at 10 kHz and sampled at 20 kHz. Fourteen Mel-frequency cepstral coefficients (MFCC) were computed every 10 ms on a 20 ms analysis window. Some of the training data was used with a nonuniform binary clustering algorithm to produce 64 representative MFCC vectors. Each MFCC vector in the training and test sets was then classified using vector quantization (VQ) [11] as one of the 64 vectors.

Recognition experiments were performed using three different models: context-independent (phoneme), left-context only, and a combined model. For each case, the system was alternately trained with 1, 4, 10, and 20 tokens per letter. The recognition was performed using a best-first stack search.

Figure 5 shows the results of the three experiments. The horizontal axis represents the amount of training speech used (tokens per letter). The circles show the performance where only phoneme models are used. In this case, with one token of each letter, there are 9 tokens of the phoneme /i/. The squares show the performance where a separate model is used for each left context. For one token of each letter, there is only one token of each context-dependent model of /i/. As can be seen, with only 1 token per letter, the left-context model performs poorly (61%), while the corresponding experiment using the phoneme model (with 9 tokens of /i/), achieves 79%. The poor results of the left-context model can be attributed to the facts that, with 1 token, it is impossible to estimate a 64-bin discrete pdf, and the model for the last part of /i/ is not very dependent on the phoneme to the left. When the number of training samples is increased to 4 or 10 tokens per letter, the performance using the left-context model improves rapidly to 88%, while the phoneme model performance improves to 93%. As more training is made available, the phoneme model performance doesn't improve (presumably because 90 tokens for each pdf in /i/ is more than sufficient training). However, the left-context model performance continues to improve to 97% with 20 tokens per letter.

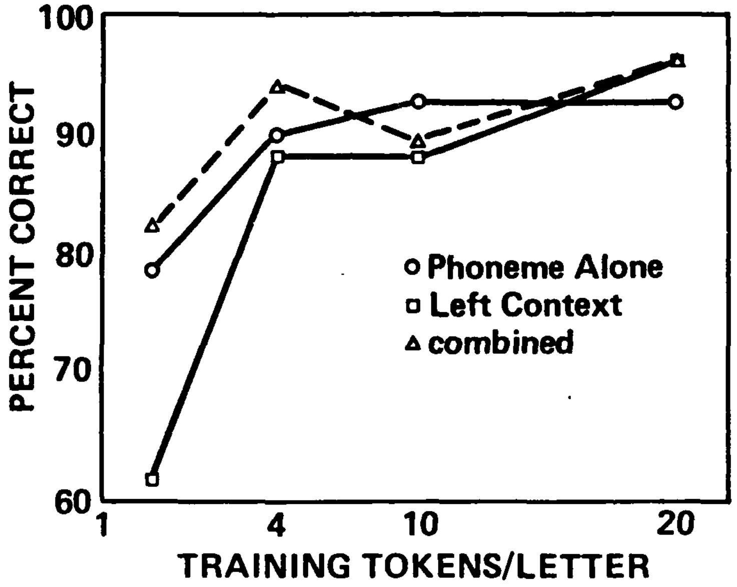
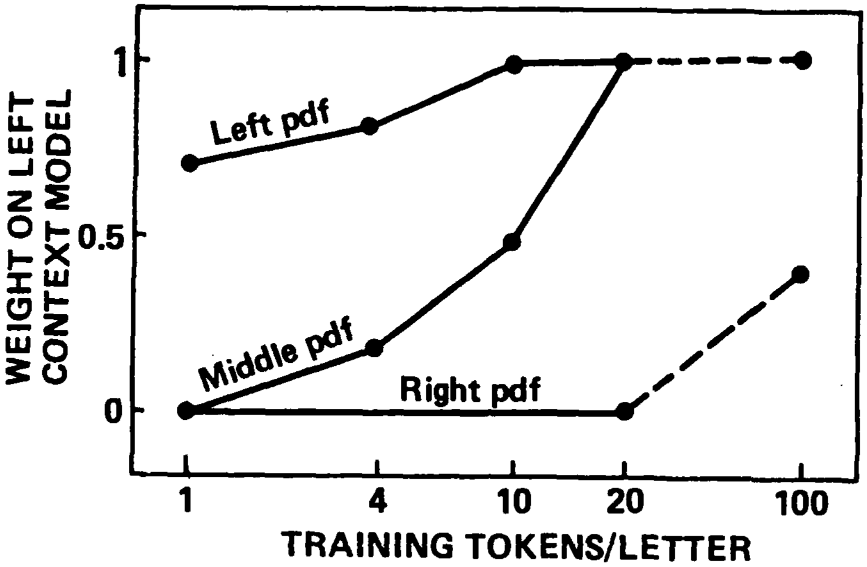


Figure 5: Combining Models: Percent correct recognition performance of E-set for 1, 4, 10, 20 training tokens per letter, for phoneme (O), left-context (□), and combined (Δ) models.

From these results, one could devise a simple algorithm that used the left-context model only for those cases where there were more than 10 tokens of the appropriate context. For the remaining cases, only the phoneme model would be used. If, as discussed in the preceding section, we allow the first part of the phoneme to depend more on the left-context model, and the other parts to depend mostly on the phoneme model, the algorithm would require fewer tokens for the context-dependent model of the pdf nearest the phoneme transition, and require more for the pdf farthest from the transition.

Finally, we consider the case where continuous weighting factors are used to combine the two models. The weights (which were set by hand) are dependent on the amount of training as well as the location within the phoneme, as shown in Figure 6. The recognition performance for the combined model is indicated in Figure 5 by the triangles. With 1 token per letter, the left-context model cannot add significantly to the performance. With 4 tokens, however, the combined model outperforms either model. With 10 tokens, the performance has dropped to below that of the phoneme model alone. It appears that in this small test, the 10 tokens of training did not produce a good model for test data. With 20 tokens, there is sufficient training for the left-context model and, therefore, smoothing with the phoneme model does not help. In general, the combined context model gives the same or better performance than either model by itself.



$$p\,(\mathrm{Ac} \mid b\,[i]\,,s) \approx$$
$$\lambda\,(\text{Left context}, s, \text{Nocc})\; p\,(\mathrm{Ac} \mid b\,[i]\,,s)$$
$$+\; \lambda\,(\text{Phoneme}, s, \text{Nocc})\; p\,(\mathrm{Ac} \mid [i]\,,s)$$
$$s = \{\text{Left, Middle, Right}\}$$

Figure 6: Weight on left-context model (vs. context-independent phoneme model) as a function of the state and number of occurrences of training.

5. Continuous Phonetic Recognition

In this section, we describe experiments on continuous phonetic recognition, using the same techniques for modeling coarticulation. The analysis methods were the same as for the previous experiments with the following exceptions. The clustering and vector quantization of the speech used several different size codebooks, from 64 to 512. A simple variable-frame-rate (VFR) algorithm was used to reduce the computation somewhat. Strings of up to 3 identical vector codes were compressed to 1 observation. (This simple variable frame rate scheme was found not to affect performance.)

Time-Synchronous Search

The recognition process requires that we find the most likely sequence of HMMs for the observed spectral sequence. To perform this search in an optimal manner requires computation that is exponential in the number of models and in time. The best-first stack search is commonly used to attempt to find the best path, with minimal computation. This strategy uses a list or stack (usually a tree) of theories for different model sequences. Then, given an evaluation criterion, it advances the most promising theory by all possible next phonemes until the best theory spans the utterance. Complex ad hoc theory-merging and theory-normalization procedures must be used to properly weigh the score and the length of the different theories to trade off between exponential search and errorful search. In the case of phoneme recognition (where the recognition units are relatively short), the overhead incurred requires orders of magnitude more computation than the simple score computation. Finally, since the decision of which theory to pursue next is inherently sequential, it is difficult to get large improvements in speed by the use of large numbers of multiple processors.

We propose, here, a new time-synchronous approximate search for the most likely sequence of hidden Markov models. First, we consider that all the phonetic models are connected in one large HMM for all speech. The well-known Viterbi algorithm finds the most likely single path through a HMM, by a simple iterative process. That is, at each time, one finds the best path to each state at that time, by finding the best path from all preceding states. We have modified this algorithm to compute, more nearly, the most likely sequence of submodels (phoneme models here). As with the Viterbi algorithm, each state is updated for each time frame. However, some scores (probabilities) propagate among states within a submodel, while others propagate from the terminal state of one model to the initial state of several other models. Whenever two or more paths come together within a phoneme model their probabilities are *added*. However, when scores from two or more different phoneme models propagate to the same other phoneme model, only the *best* score and path is remembered. This dynamic programming process is performed until the end of an utterance. Then, the best sequence of phoneme models through the large HMM is determined. In principle, this approach considers all possible sequences of models.

This approximate solution has been found to result in somewhat higher recognition accuracy than the Viterbi algorithm, while avoiding the computation and complexity of the best-first stack search. When this algorithm is augmented by a simple beam-search technique, which prunes out all theories sufficiently below the best theory, the computation can be reduced by orders of magnitude further, with only a small probability of getting a different answer than the exhaustive search. Since the recognition is performed left to right, the delay from the end of an utterance until the answer is found is small and fixed. Finally, we have shown in another project [12] that this algorithm can be made to achieve near linear speedup on a Butterfly^{TM} multiprocessor with up to 100 processors operating in parallel.

Database

The speech training database for this experiment consisted of 25 minutes of speech, containing a total of 550 sentences spoken by a single talker, covering three different topics: office type queries (budgets, messages, trips, etc.), Harvard sentences, and children's books. One hundred sentences were digitized in each of several recording sessions spaced several days apart. The training and additional test data were later transcribed.

Figures 7a and 7b compare the phonetic recognition performance for several different configurations of the system. The accuracy numbers shown in the figure were computed as the percentage of phonemes in the speech that were found in the output phoneme string in the correct order. Thus, the measure takes into account both substitutions and deletions, but not insertions. The number of insertions was typically around 12% for all experiments.

In general, the models that are derived from a combination of the phoneme model and either the left or right context-dependent model resulted in significantly better performance than either the context-independent phoneme model or the left-context model alone. The system that used a combination of models dependent on left and right context simultaneously did not improve performance any further. A careful examination of the results showed that including either left context or right context produced similar answers.

As can be seen in Figure 7a, the performance improves with a finer spectral resolution in the VQ codebook, as long as the training set is sufficiently large. With only five minutes of training, performance improved as the number of spectral templates increased from 64 to 256. However, for the PH+L case, the performance dropped when the number increased to 512 spectral templates, presumably due to insufficient training data for each pdf. As shown in Figure 7b, as the amount of training was increased to 25 minutes, the performance improved most for

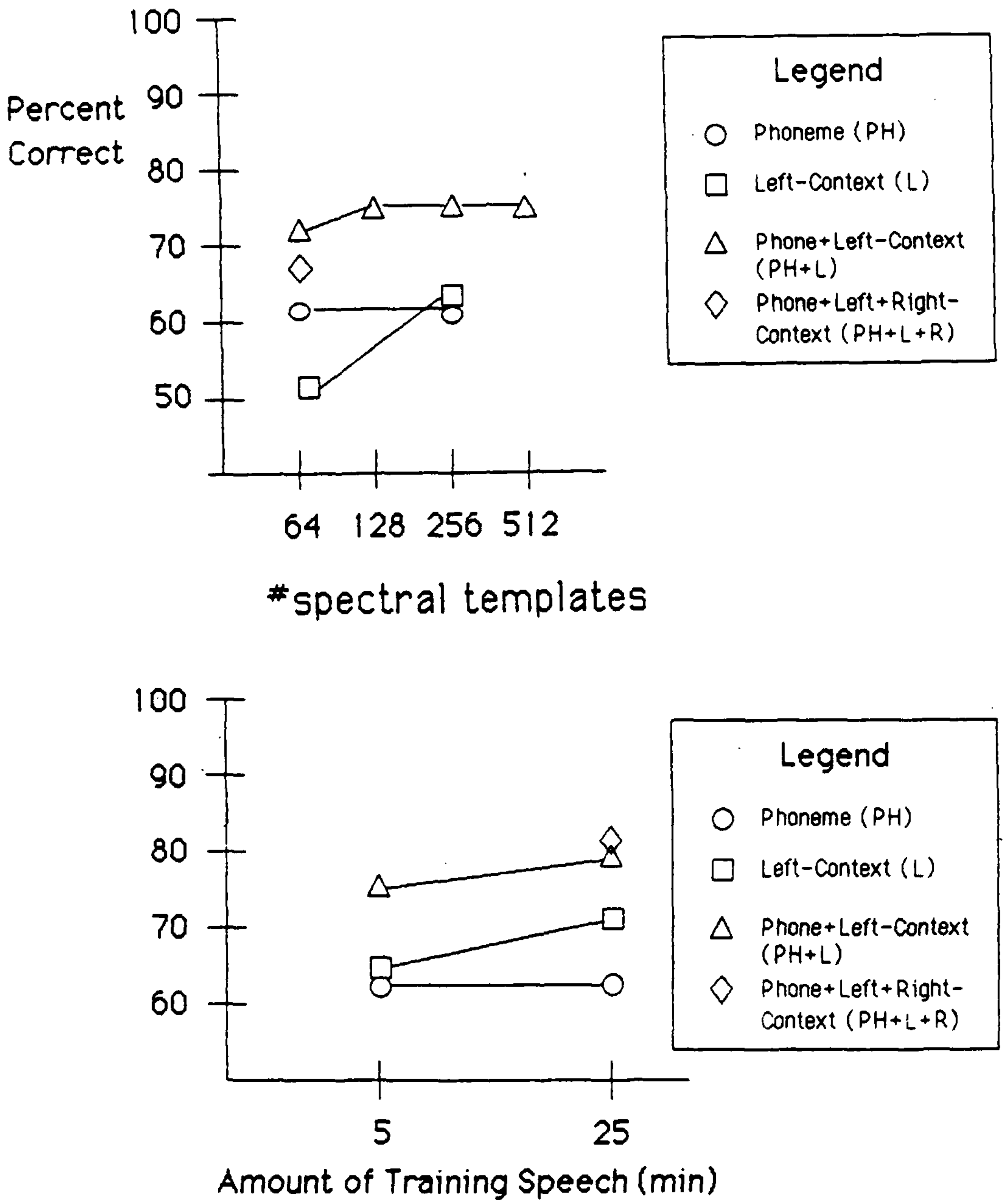


Figure 7: Phonetic recognition accuracy for continuous speech using different context models. a) Accuracy vs spectral codebook size; b) Accuracy vs amount of training speech.

those systems that used combined models and a large number of spectra. In particular, the combined phoneme+left+right context model, with 256 spectra, and 25 minutes training cut the errors in half (81% correct) relative to the context-independent (phoneme) model alone (62%).

6. Word-Dependent Coarticulatory Effects

In this section, we extend the coarticulation model to the problem of continuous speech, large vocabulary word recognition. In our phonetic recognition experiments we have observed that the improvement in performance due to using left- or right-dependent models of phonemes instead of context-independent models is smaller when the test vocabulary is different from the training vocabulary, even though the contexts in the test set had occurred frequently in the training set. We hypothesized that contexts beyond the immediate phonetic contexts are important and affect recognition results. This might explain why speech recognition systems that model whole words typically outperform those that use a phoneme model, as long as the amount of training for each word is sufficient and the effects between words are not severe. However, word-based systems cannot easily take into account word boundary effects and are not easily extensible to vocabularies of thousands of words. The problem then is to model phonemes in context to maximize recognition performance on a particular large vocabulary, especially when not all the words in the vocabulary appear often enough in the training set to allow the estimation of robust models.

To extend our model of coarticulation to the word level, we need only include a word-dependent model of the phoneme with any other models that we choose to use. We also must expand our dictionary pronunciations to permit modeling of the desired context. For example, Figure 8a shows a phoneme network containing the alternate pronunciations for the word "data". If we want to be able to represent triphone (joint left and right) contexts, we must expand the network as shown in Figure 8b. In Figure 8b each of the pronunciations is labeled with a phoneme and its left and right context.

Database

The vocabulary used in this study was from a 334-word electronic mail task. The task has a fairly rich structure and allows many different types of questions and commands, such as:

- Print all messages from Smith on the Dover.
- Which messages have I deleted since yesterday?
- Has Jones replied to my last message?

A total of 400 different sentences were generated covering 250 words of the vocabulary. The sentences were each recorded by three male speakers and one female speaker in sessions of 100 sentences, separated by a few days. The first three sessions were designated as training data, and the last as test material. The total duration of the training material was thus about 15

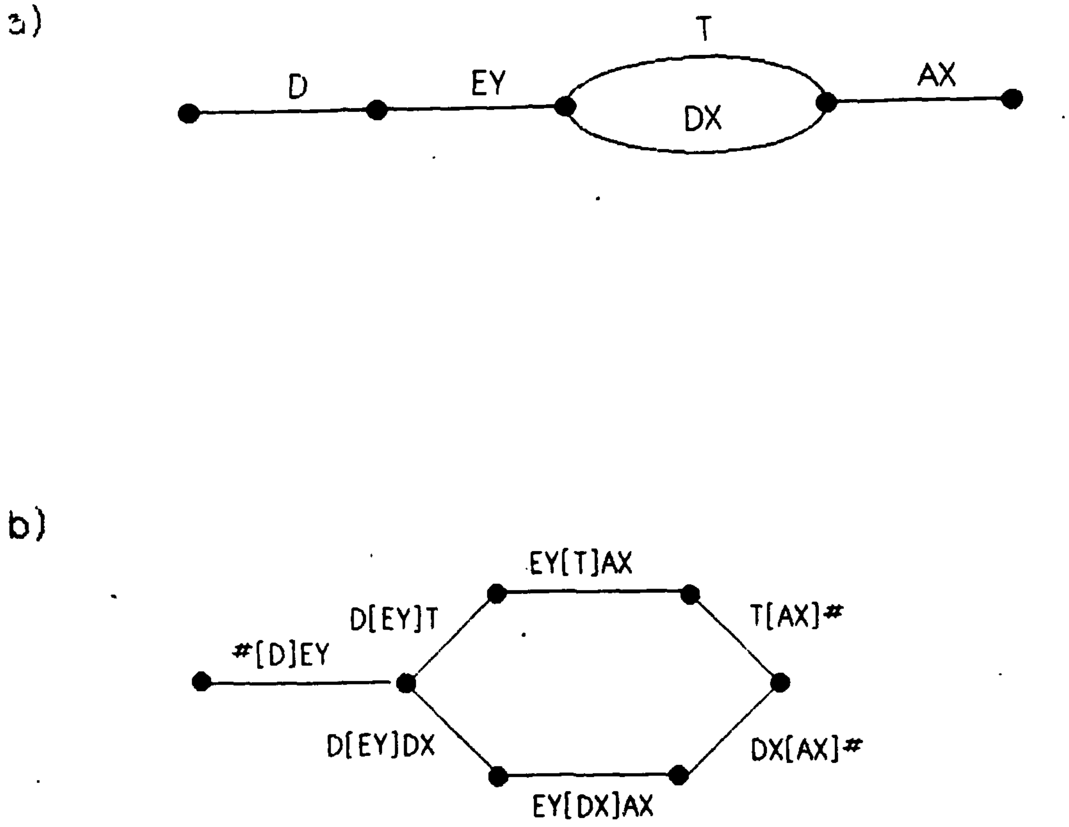


Figure 8: Phonetic Context Dictionary Expansion: a: Original phonetic network; b: Expanded triphone network for a word.

minutes for each speaker. The test material used in the experiments below included 30 of the test sentences, with a total of 187 word tokens covering 80 different words.

A dictionary of phonetic pronunciations was constructed for this 334-word vocabulary without listening to either the training or test material, but by trying to account for the most frequent phonological variations for each word. The average number of different pronunciations per word was 2. Word boundary phonological variations were *not* included. (In a separate experiment, each word was allowed only one pronunciation. The recognition accuracy was slightly higher than with multiple pronunciations. We have not fully explained this result, and are not sure whether it will carry over to very large vocabulary experiments.)

Analysis

The sentences were read directly into a close talking microphone in a natural but deliberate style in a quiet office environment. As before, some of the training data was used with a clustering algorithm to produce a representative set of MFCC vectors. However, in this case we used a k-means clustering, which was found to result in slightly better performance than the nonuniform binary clustering procedure. These experiments were performed using a codebook size of 256 MFCC templates. We used the simple VFR algorithm described above to save computation.

Training

To obtain the necessary initial estimate for the probability density function (pdf) for each state of the phonetic HMM we use a bootstrapping technique. A separate passage (5 minutes of speech of a different vocabulary) spoken by one of the male talkers is carefully labeled, indicating the beginning frame of each phoneme. The hand-labeled speech is then quantized using the VQ codebook for each particular talker in the experiment. Normalized histograms of the observed vector-quantized spectra for each phoneme are computed from the labeled data to form an initial estimate of the pdf for the phoneme for that talker. All the pdfs for the different states in the HMM for a phoneme are set to this initial estimate. Finally, all the pdfs for the context-dependent models of a phoneme are set equal to the single, context-independent model of that phoneme. This bootstrapping technique of using a single talker's speech as an initial estimate for all talkers seems to work quite well for both male and female talkers.

We have also used a second bootstrapping technique that gives approximately the same performance without any manual labeling effort. We start from a flat initial estimate for each phoneme, and train the system using context-independent models only until convergence. Then, these models form the initial estimate for the context-dependent models, which are then trained further. This second method requires more computation, because of the need for two training sequences, but makes no assumptions about the nature of the acoustic environment, or the availability of manually labeled speech.

The 15 minutes of training data per talker is transcribed with the sequence of words spoken (no time labels and no phonetic labels). The training data is then processed with five passes of the Forward-Backward algorithm, which is normally sufficient for convergence. In the cases where context-dependent models of the phonemes are used, the training algorithm maintains separate models for each observed phonetic context. The numbers of different acoustic models found in the training set were: 50 context-independent, 500 left- or right-context dependent, and 1600 word-dependent models.

Prior to recognition, word models are precomputed for each word in the vocabulary from the appropriate phoneme-in-context models with weights depending on the number of occurrences of each model and the position within the phoneme (as used in training) as illustrated in Figure 4.

Recognition

The recognition algorithm used was the time-synchronous approximate procedure described above. No grammar was used, thus making the branching factor equal to the vocabulary size (334). The recognized sequence of words was then compared automatically to the correct answer to determine the percentage of correct, deleted and inserted words. Word substitutions and deletions are tabulated as errors, while insertions are counted separately.

We present results for several different context models. As described in the previous section, the results were produced for the following set of conditions: 3 speakers, speaker-dependent, 334-word lexicon, electronic mail task, no grammar, 15 minutes of training, and 30 test utterances totaling 187 words. Table 1 gives a detailed description of the various system configurations for the different experiments.

Figure 9 shows the word recognition accuracy for each coarticulation model (identified below the graph). The left and right axes show the percentage of words correct and percent error correspondingly. This performance measure only takes into account substitution and deletion errors. Therefore, the percentage insertion errors (i.e. the number of extra words divided by the number of words spoken) is given directly above each label. For each coarticulation model, the performance is indicated for each male speaker by a filled circle. The average performance across speakers is indicated by the horizontal line. Finally, the single triangle for system PH+W indicates the recognition performance for the female talker. For this best system, the word recognition accuracy, averaged across the four speakers, was 90%.

System Name	*Word models are constructed using:*
PH	Context-independent phoneme models
W	Only word-dependent phoneme models, regardless of whether training is sufficient for the word
PH+W	Linear interpolation of context-independent and word-dependent phoneme models
PH+L+R	Linear interpolation of context-independent, left-context-dependent and right-context-dependent phoneme models.
PH+L+R+W	Linear interpolation of context-independent, left-context-dependent, right-context-dependent, and word-dependent phoneme models.

Table 1: Different System Configurations for Word Recognition.

From the results given above, we make the following observations. First, the systems that model coarticulatory effects clearly result in better recognition performance. For example, system W achieves significantly better performance than system PH. Note that in this experiment, while not all vocabulary words were in the training set, all words in the 30 test sentences were observed at least once in the training. Although some words are poorly trained, the overall performance is improved. Note that for larger vocabularies, many words would not occur in training, making this system (W) inappropriate; a system that uses a subword context-dependent model will be necessary. Second, the systems that use less detailed models to smooth the highly context-dependent models result in higher accuracy and fewer insertions than those that attempt to use the context-dependent model by itself. For example system PH+W outperforms system W. Third, the range in performance across the three speakers (17%) is large for the context-independent (PH) system. We conjecture that this is due to a difference in the degree of coarticulation present. However, the range in performance for the context-dependent systems (4-6%) is greatly reduced - a desirable attribute. We believe this behavior is due to the fact that these systems are better able to model the coarticulation present.

As a side note, we tried combining all four models (PH+L+R+W) in a single experiment, but found that performance did not improve over the PH+W system. We presume that this is due to the fact that most words in the test set were well trained.

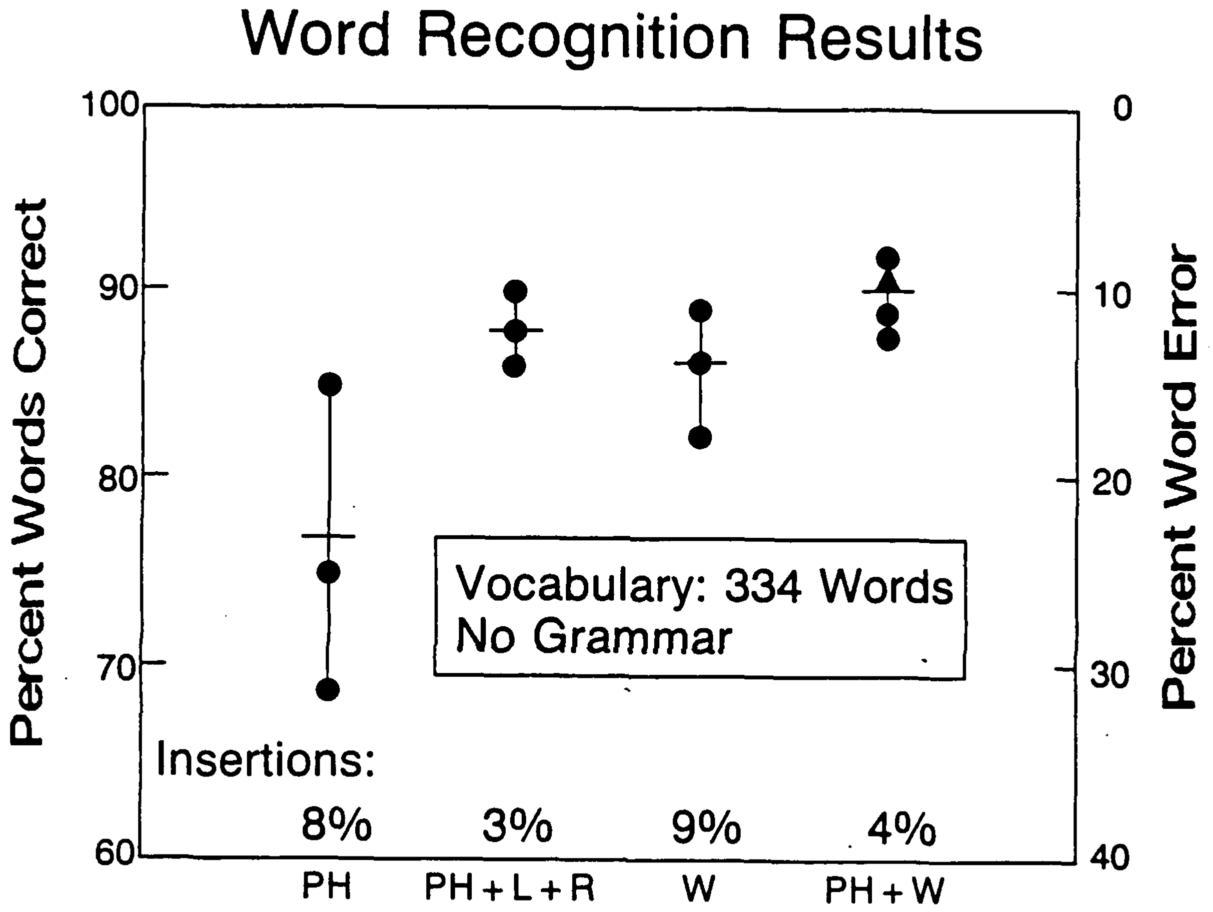


Figure 9: Word Recognition Accuracy using different coarticulation models.

7. Recognition With a Grammar

Our next goal was to use the coarticulation model in a complete continuous speech recognition system. We constructed a deterministic grammar for the electronic mail task using an extended context free notation. The rules were compiled into a finite-state network. To do this, we disallowed infinite recursion. The resulting network had approximately 600 nodes and 4000 arcs. Each arc represents a word and each path through the network represents a valid sentence in the language. We measured the complexity of the network in two ways: Maximum Perplexity, and Test Set Perplexity. The Maximum Perplexity, is derived by finding the number of sentences of each length in the language. From this, [13] shows how to find the maximum value for the perplexity of the language. The maximum perplexity of the grammar network used was 60. This measure is probably an overestimate of the difficulty of the problem, since any particular test set will not likely have the distributions specified by the maximum perplexity probability assignments. To derive a more realistic estimate of the complexity, we measure the Test Set Perplexity on an independent set of sentences [4]. First, each sentence in a test set is parsed by the grammar. Then we compute the geometric mean of the number of possible words at each node of the grammar, sampled over the test set. The Test Set Perplexity was 31. While the perplexity of the a language does not take into account the acoustic confusability of the competing words, we feel that the Test Set Perplexity measure is still a good *rough* measure of task difficulty.

The recognition algorithm used was the same time-synchronous search, with the modification that each word-arc could only be followed by those word-arcs allowed by the grammar. While the computation for a large grammar would increase proportionally with the number of arcs in the grammar, we found that it was possible to prune most of the paths using a beam search, without any loss in performance. In fact, the recognition was typically much faster than when no grammar was used.

Table 2 shows the recognition accuracy for coarticulation models PH and PH+W. The table gives both the word accuracy (percentage of words correctly recognized) and the sentence accuracy (percentage of sentences recognized exactly correct with no insertions, substitutions, or deletions). The results are averaged across the 3 male and 1 female speakers. As seen, system PH+W, has about one fourth the word errors, and less than one third the sentence errors of system PH. The word recognition accuracy with a grammar was 98.8%, averaged over the four speakers. The sentence recognition accuracy was over 90%.

Context	*Word Accuracy*	*Sentence Accuracy*
PH	94.7%	66.4%
PH+W	98.8%	90.2%

Table 2: Continuous speech recognition results.

8. Conclusion

We have presented a formalism for modeling coarticulatory effects in a robust way. The formalism uses detailed context-dependent models of phonemes smoothed by more robust context-independent models, with weights that depend on the amount of training of each model and the location within the phoneme. Thus, the phonetic modeling in the recognition system is not tied to any particular level of context, such as the diphone or syllable. It attempts to use the information in the training data to the extent possible. Experiments have been performed on four different tasks: Isolated E-set recognition, continuous phonetic recognition, continuous word recognition, and continuous speech recognition using a grammar. The speaker-dependent recognition accuracies for these four problems were: E-set: 97%, phoneme recognition: 81%, continuous speech word recognition (no grammar): 90%, and continuous speech grammar recognition (text-set perplexity of 31): 98.8%. In all cases, the benefit of using the robust coarticulation model over the simple context-independent phonetic model was a reduction of the error rate by at least a factor of two and often more.

9. Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency and was monitored by the Office of Naval Research under Contract No. N00014-85-C-0279.

The authors wish to thank Alex Rudnicky of CMU for supplying us with the initial set of words and sentences for the electronic mail task.

10. References

1. S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, April 1983, pp. 1049-1052.

2. J.K. Baker, "Stochastic Modeling for Automatic Speech Understanding", in *Speech Recognition*, Raj Reddy, ed., Academic Press, New York, 1975, pp. 521-542, ch. Part Five:systems Organization and Analysis Systems.

3. L.R. Bahl and F. Jelinek, "Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition", *IEEE Trans. Inform. Theory*, Vol. IT-21, No. 4, July 1975, pp. 404-411.

4. L.R. Bahl, F. Jelinek, and R.L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, March 1983, pp. 179-190.

5. L.E. Baum and J.A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model of Ecology", *Amer. Math Soc. Bulletin*, Vol. 73, 1967, pp. 360-362.

6. N.R. Dixon and H.F. Silverman, "The 1976 Modular Acoustic Processor (MAP)", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-25, No. 5, October 1977, pp. 367-379.

7. R.M. Schwartz, J. Klovstad, J. Makhoul, and J. Sorensen, "A Preliminary Design of a Phonetic Vocoder Based on a Diphone Model", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, April 1980, pp. 32-35.

8. A.E. Rosenberg, L.R. Rabiner, S.E. Levinson, and J.G. Wilpon, "A Preliminary Study on the Use of Demisyllables in Automatic Speech Recognition", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, April 1981, pp. 967-970.

9. O. Fujimura, "The Syllable as a Unit of Speech Recognition", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-23, No. 1, February 1975, pp. 82-87.

10. M.J. Hunt, M. Lennig, and P. Mermelstein, "Experiments in Syllable-Based Recognition of Continuous Speech", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, April 1980, pp. 880-883.

11. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding", *Proc. IEEE*, Vol. 73, No. 11, November 1985, pp. 1551-1588, Special Issue on Man-Machine Speech Communication.

12. L. Cosell, O. Kimball, R. Schwartz, and M. Krasner, "Continuous Speech Recognition on a Butterfly^{TM} Parallel Procesor", *Int. Conf. on Parallel Processing*, St. Charles, IL, August 1986.

13. M.M. Sondhi and S.E. Levinson, "Computing Relative Redundancy to Measure

Grammatical Constraint in Speech Recognition Tasks'', *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tulsa, OK, April 1978, pp. 409-412.

END

12-86

DTIC